

StrEmbed-4 Users' Manual

Embedding design structures in engineering information

Hau Hing Chau

13 July 2017

Version 4E

What is StrEmbed-4?

Structure Embedding version 4 (StrEmbed-4) is a deliverable from the Embedding Design Structures in Engineering Information project, a Design The Future project funded by Engineering and Physical Sciences Research Council (grant reference EP/N005694/1).

Engineers use design structures, such as Bills of Materials (BoMs), to tailor product definitions, including shape, for particular activities. For example, an engineering BoM defines the as-designed product whereas a manufacturing BoM defines the as-built state of the same product and a service BoM includes information on how the product has been maintained. All of these BoMs relate to the same designed product. However in practice, because of restrictions arising from current computer aided design technologies and associated business systems, different BoMs are usually related to separate digital definitions of the same product. This creates significant data management problems that add cost, time and rework into product development processes. If resolved, substantial business benefits, through improved efficiency and effectiveness of product development processes, could be achieved.

StrEmbed-4 demonstrates how a design structure, for example “as designed”, typically using a computer aided design (CAD) system can be embedded on to a corresponding hypercube lattice. This embedding relationship can be shown visually with this software. Any combinations or permutations of an alternative design structure is also embedded in this hypercube lattice. Modified design structure can be exported to a data format that can be read by a CAD system. A lattice is a partially ordered set (poset) which is equivalent to a corresponding universal algebra. This algebra has a number of useful mathematical properties that allow generation of alternative design structures for other stages of product life cycle, *inter alia*, “as manufactured” and “as maintained”.

Installation

Operating system

StrEmbed-4, in principle, could be run on any 64-bit operating system. It is tested on Microsoft Windows 7 for x64-based systems, and Debian GNU Linux version 6.0 Squeeze amd64 port. If you use another operating system and have difficulty running StrEmbed-4, please contact the author.

Perl 5 scripting language

StrEmbed-4 is written in Perl 5. It works on Perl version 5.12.0 or later, but it is not compatible with Perl 6 which has an entirely different code base.

Perl modules

StrEmbed-4 uses a number of Perl [library] modules. There are required on the top of a Perl 5 base installation. They are listed below.

<i>Module name</i>	<i>Package name</i>	<i>Description</i>
Tk	Tk	Tk – a Graphical User Interface Toolkit
Set::Partition	Set-Partition	Enumerate all arrangements of a set in fixed subsets

Installing Perl 5 and its modules on Windows

Perl 5 – There are many flavours of Perl. Here we provide an example of installing ActivePerl <<http://www.activestate.com/activeperl/downloads>> on Windows. Perl 5 can be obtained from the following. Do follow vendor's on-screen instructions.

A link for Windows Installer (EXE) for ActivePerl version 5.24.0.2400 for Windows (64-bit, x64) is listed below. Please check Active State web site for the latest version <<http://www.activestate.com/activeperl/downloads>>
<http://www.activestate.com/activeperl/downloads/thank-you?dl=http://downloads.activestate.com/ActivePerl/releases/5.24.0.2400/ActivePerl-5.24.0.2400-MSWin32-x64-300558.exe>

StrEmbed-4 works with Perl 5 but not Perl 6.

Perl modules – ActivePerl provides a graphical Perl Package Manager.

- 1/ Entry package name(s) in the search box.
- 2/ Click "Mark for install [+]". This button is located at the right hand side of the search box.
- 3/ Select all required Perl modules listed in the above table.
- 4/ Click "Run marked actions [Ctrl-Enter]".
- 5/ Installation will take 5-10 minutes. Check "Status" window for further instructions, if necessary.

Other flavours of Perl 5 on other operation systems

Comprehensive Perl Archive Network (CPAN) <<https://www.perl.org/cpan.html>> provides Perl for a large number of computing platforms. Perl modules are organised using a command line perl package manager (ppm). There is a little learning curve for it. Nonetheless, all necessary instructions are available from CPAN.

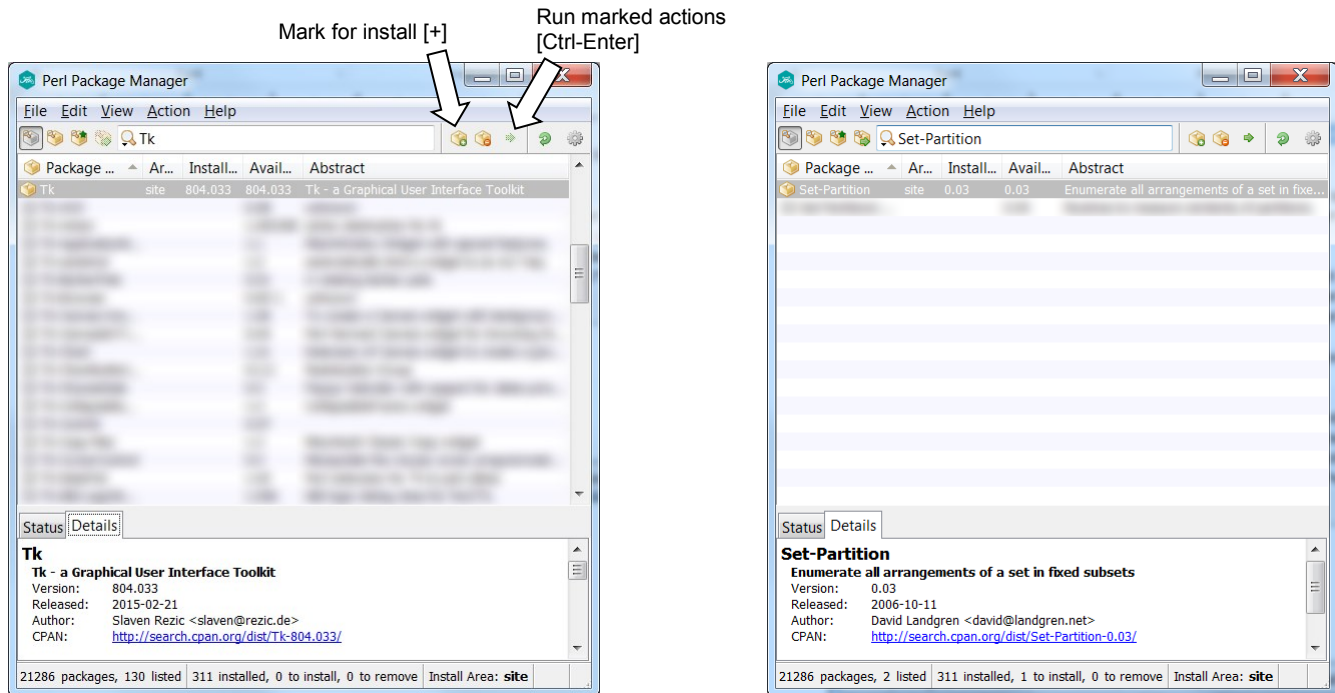
Strawberry Perl <<http://strawberryperl.com/>> is another popular flavour of Perl. One also requires to use command line ppm to manage Perl modules.

Running StrEmbed-4

In a Windows Explorer, double click `StrEmbed-4.pl` Alternatively, at a Windows command prompt (cmd.exe), in sub-directory `bin` type `perl StrEmbed-4.pl`

On a Unix (including Linux, Mac OSX and macOS), at a console, in sub-directory `bin` type `chmod a+x StrEmbed-4.pl` and follow by `./StrEmbed-4.pl`

Strawberry Perl <http://strawberryperl.com/> is another popular flavour of Perl. One also requires to use command line ppm to manage Perl modules.



A Walk through example

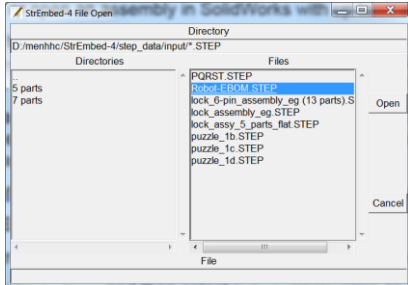
Export a STEP AP214 file from a SolidWorks assembly

- 1/ Create or open an assembly in SolidWorks with up to seven parts.
- 2/ Export the assembly to a STEP AP214 file by **File** → **Save As...**, in **Save as type:** box select **STEP AP214 (*.step;*.stp)**, and then click **Save** button. Do make sure file extension is **.STEP** all caps which is the default setting. If a different file extension is used, the file needs to be renamed before you proceed to the next step.
- 3/ There are limitations on input STEP files: (i) no more than seven parts, (ii) no special characters for part or assembly name, especially no full stop ".", (iii) each part is used for one and only one instance; and (iii) each sub-assembly is consisted of at least two parts and/or sub-assemblies.
- 4/ Alternatively, instead of creating one's own STEP file, one could skip to Step 5 and use a pre-prepared STEP file.

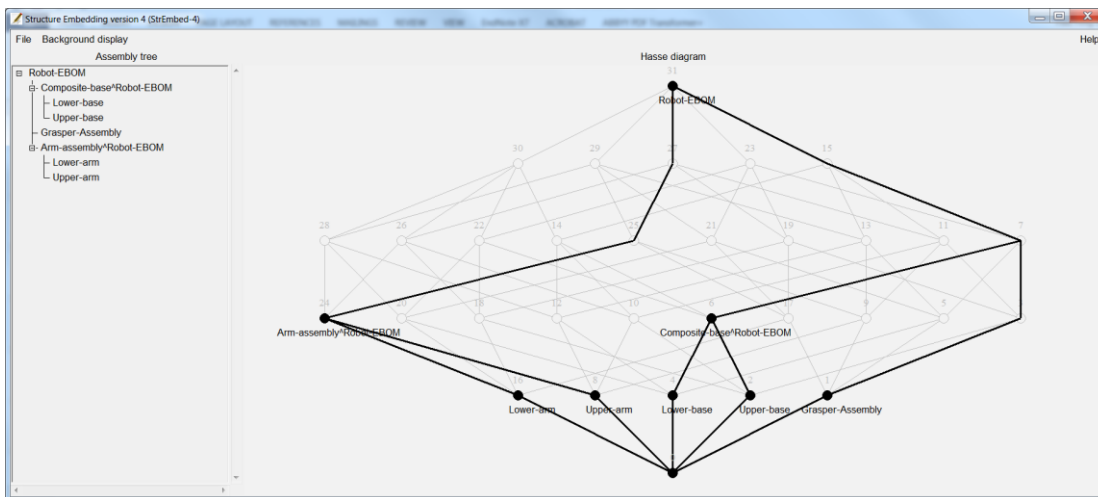
Read a STEP file using StrEmbed-4

- 5/ Run StrEmbed-4 by double click `StrEmbed-4.pl` or follow the instructions in the previous sections.

- 6/ Click File > Open and then select Robot-EBOM.STEP or one's own STEP file. Default file location is step_data/input/ but one may navigate to any other sub-directories as required. Do note that only files with an extension of .STEP all caps are visible.



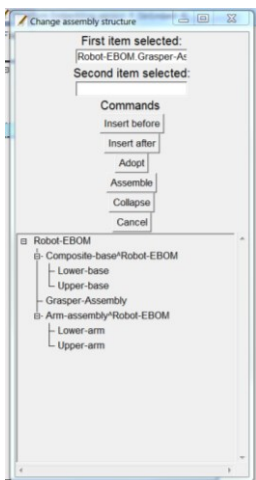
- 7/ After a STEP file is loaded, a corresponding assembly tree is displayed on the left and a Hasse diagram on the right.



- 8/ Unused hypercube lattice elements, which are greyed out, could be toggled On or Off by selecting corresponding options under Background display on the menu bar.

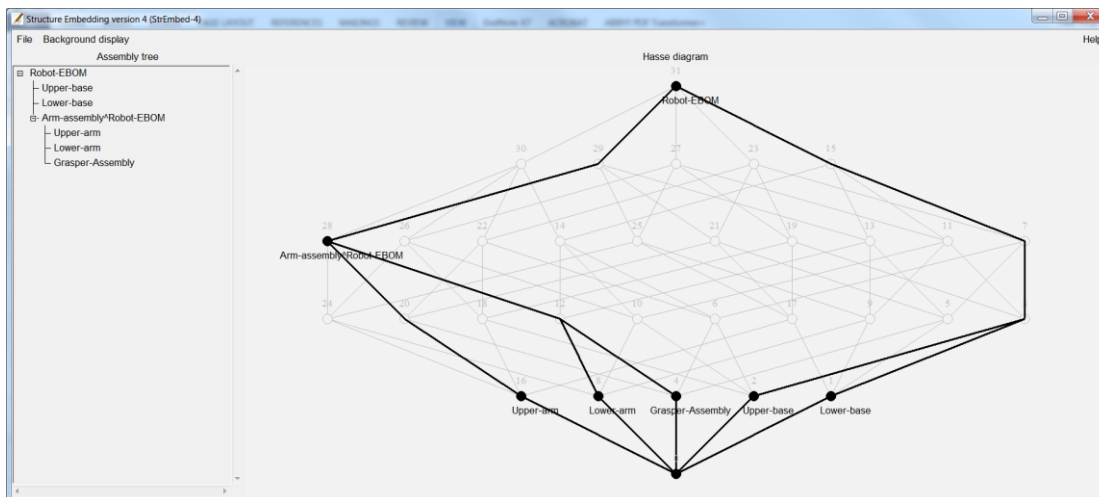
Change product structure

- 9/ Double click assembly structure and an editor window will pop up.



- 10/ There are four commands available for use to modify the existing assembly structure: (i) insert before or insert after, (ii) adopt, (iii) assemble; and (iv) collapse.
- 10a/ At most two items could be selected for each command operation. Click and hold mouse cursor to select the first item. Continue to hold and to drag, and release mouse button over the second item.
- 10b/ *Insert before* reorders the first item preceding the second. *Insert after* reorders the first item succeeding the second. Both items must be within the same sub-assembly.
- 10c/ *Adopt* reparents the first item to the second. The second item, which is the new parent of the first, must be an assembly but not an atomic part.
- 10d/ *Assemble* constructs a new sub-assembly with the two selected items.
- 10e/ *Collapse* removes a sub-assembly consists of only one part. This part is reparented to its original grandparent. An assembly with a single part is not allowed within StrEmbed-4.
- 10f/ Click *Cancel* or move mouse cursor outside the popup window to cancel current operation and returning to the main window.

Save a STEP AP214 file with a new product structure



- 11/ Click *File > Save* to save a new STEP AP214 file that consists of the modified assembly structure. Default file location is *step_data/output/* but one may navigate to any other sub-directories as required. Default file name is *original_modified.STEP* but one may change it as required.
- 12/ Click *File > Exit* to quit the programme.

Known bugs

- 13/ StrEmbed-4 does not support sub-assembly with a single part. If that happens, a sub-assembly with its sole part will collapse into one single part silently without warning.
- 14/ StrEmbed-4 supports assembly with up to seven parts, corresponding and appropriate sub-assemblies, and one top level assembly.
- 15/ Importing and exporting of STEP AP214 was tested on SolidWorks 2015. In principle, these functions should work correctly on other CAD systems. Please report any problems to the author. We use STEP AP214 (ISO 10303-214:2010), and aware that it was superseded by ISO 10303-242:2014. We do not expect any practical difference within the limited scope of application protocol AP214 that we used. If you believe the contrary, we would like to hear from you.

16/ StrEmbed-4 is a proof-of-concept prototype. It does its job well when one goes by the script. However, it does not handle exceptions well nor has anywhere near enough useful error messages.

17a/ Note: StrEmbed-4 has a modular design and is meant to be reused. Different modules have minimal interactions. The three main modules are (i) for **lattices** and posets, especially producing lattice representation conforms to LatDraw <<http://latdraw.org/>> by Ralph Freese from the University of Hawaii, and also among other things calculating meets and joins, (ii) for importing, interpreting and exporting **STEP** AP214 files; and (iii) for a graphical user interface (**gui**) using Perl/Tk.

17b/ More note: StrEmbed-4 is released under GNU GPL 3.0. Therefore, you can incorporate it in your own work as long as you give appropriate acknowledgements, make your source code available to others; and do not incorporate it into proprietary applications without seeking further permission from us.

People

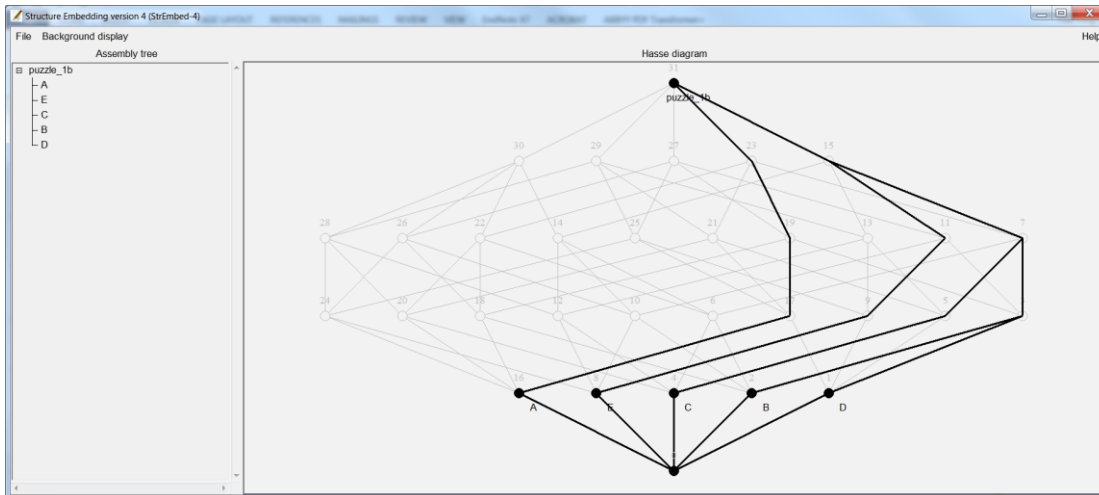
The Design Structures in Engineering Information (Embedding) project is jointly hosted by the University of Leeds and The Open University. Members of the Embedding project are Amar Behera, Hau Hing Chau, Chris Earl, David Hogg, Alison McKay, Alan de Pennington and Mark Robinson.

Getting help and reporting bugs

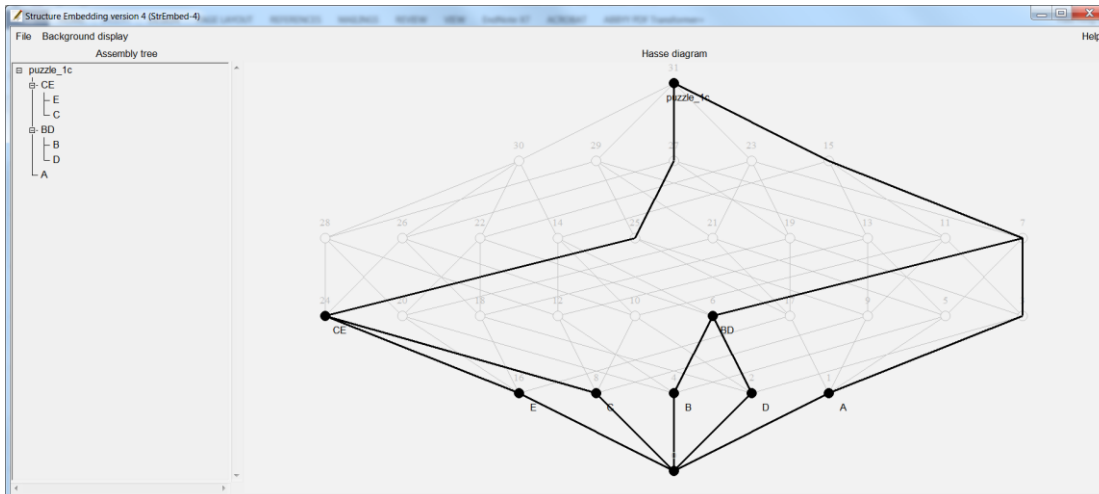
Send help request and bug report to Hau Hing Chau <H.H.Chau@leeds.ac.uk> School of Mechanical Engineering, University of Leeds, Leeds, LS2 9JT, UK.

Example screenshots

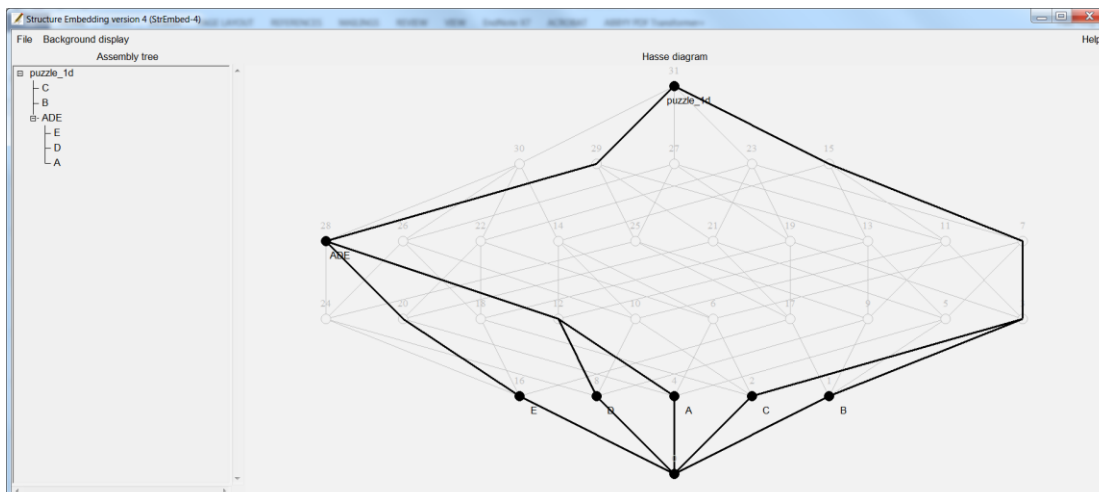
puzzle_1b.STEP



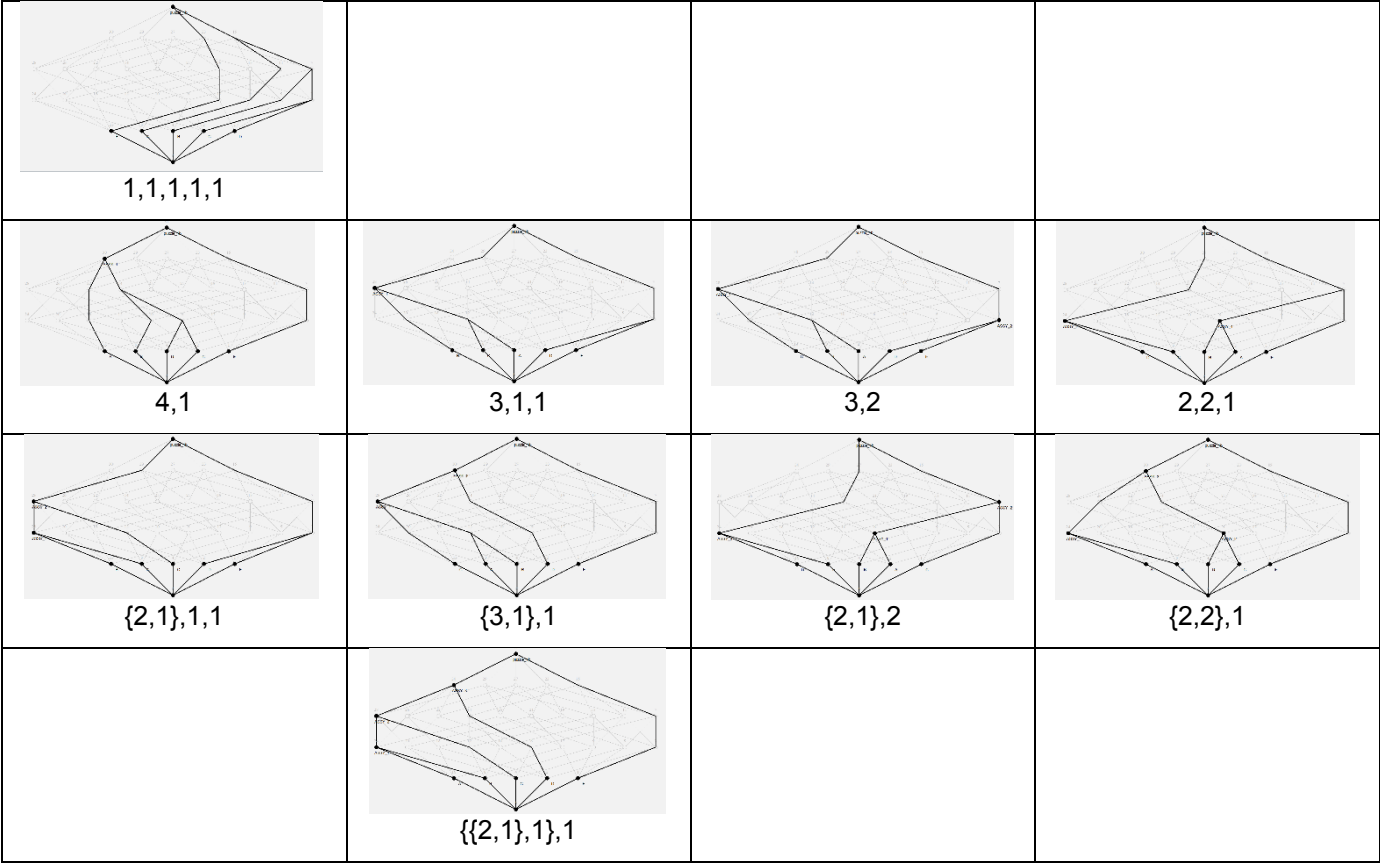
puzzle_1c.STEP



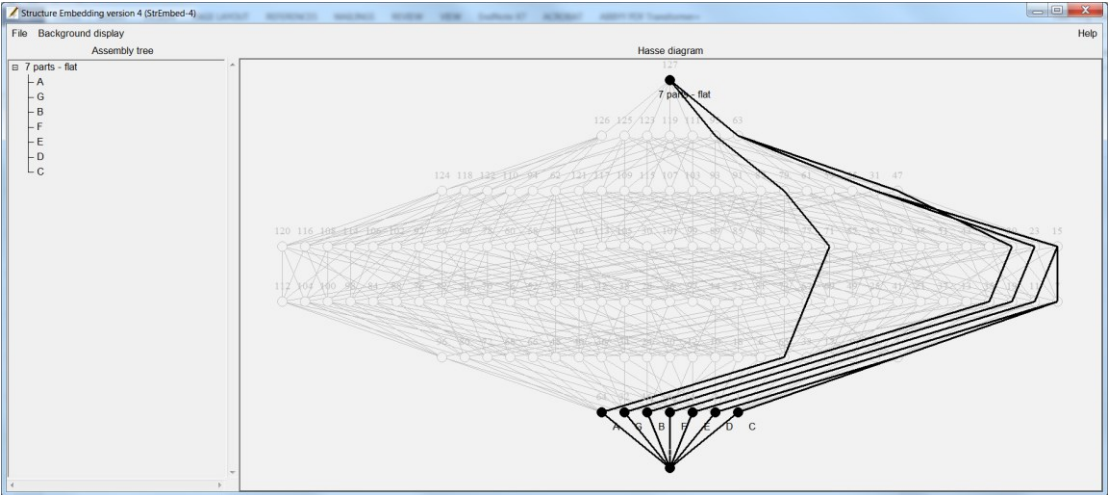
puzzle_1d.STEP



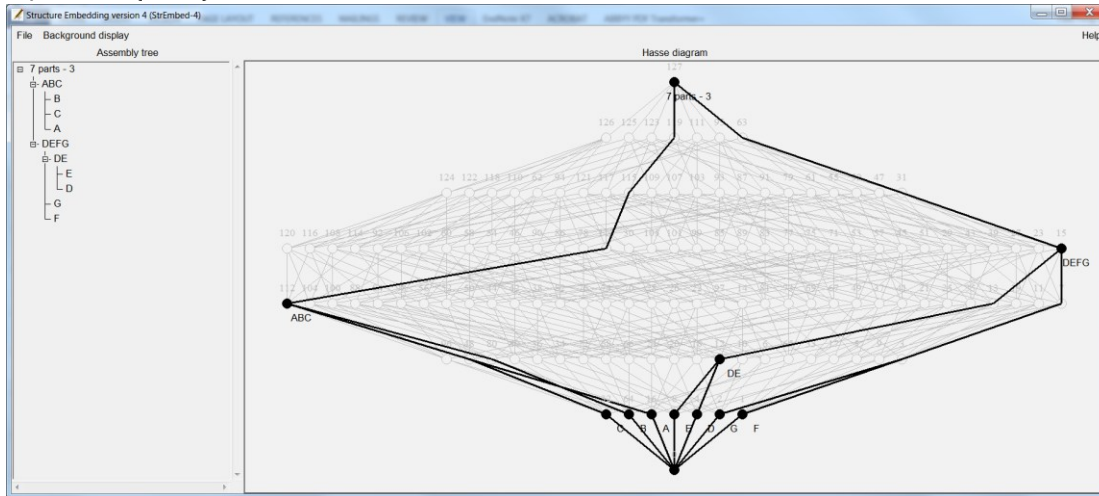
All ten possible assembly sturctures for five atoms



7 parts - flat.STEP



7 parts - 3,{2,1,1}.STEP



An assembly with 13 parts is beyond the limit of what StrEmbed-4 could visualise.

